УДК 004.93'1;004.932

Д.А. Гончаров, А.М. Недзьведь, С.В. Абламейко
Объединенный институт проблем информатики НАН РБ, г. Минск, Беларусь
goncharovda@gmail.com, nedzveda@newman.bas-net.by

Анализ методов построения объемных дистанционных карт*

В статье рассмотрены и сопоставлены друг с другом по нескольким аспектам (точность, скорость, простота реализации, возможность распараллеливания) методы построения трехмерных дистанционных карт, в результате чего можно с определенной долей уверенности дать ответ на вопрос, в каком случае лучше использовать тот или иной метод.

Введение

Дистанционная карта (дистанционное преобразование) – это такое представление данных, в котором для каждой точки объекта известно расстояние до ближайшей точки фона. Впервые в литературе по обработке изображений дистанционные преобразования появились в 1966 году в работе Розенфельда и Пфалца [1], в которой они применили дистанционное преобразование Чамфера для построения скелета объекта. С тех пор был не только улучшен тот алгоритм, но и были придуманы альтернативные решения. Были разработаны алгоритмы для различных представлений данных: шестигранных сеток, точечных решеток, продолговатых сеток. Дистанционные преобразования применяются во многих областях обработки изображений (морфологическая обработка [2], скелетонизация и построение срединных осей [3]), визуализации (ускорение метода бегущего луча, эндоскопия), моделировании (обнаружение столкновений, морфинг, анимация, конструктивная блочная геометрия). В связи с увеличивающимся применением трехмерных объектов в различных областях науки, возникла необходимость в быстрых и точных алгоритмах для построения объемных дистанционных карт.

Целью данной работы является систематизация известных алгоритмов построения дистанционных карт, выявление наиболее быстрых и точных из них, а также оценка области дальнейшего применения каждого из них.

Классификация алгоритмов

Алгоритмы для построения карт расстояний могут быть классифицированы по нескольким критериям.

Классификация по способу вычисления расстояния:

– Дистанционные преобразования Чамфера, где новое значение расстояния для вокселя вычисляется из расстояний его соседей и соответствующих весов маски.

– Векторные дистанционные преобразования, в которых для каждого обработанного вокселя хранится вектор ближайших точек поверхности, а для необработанного вокселя этот вектор строится, используя вектора соседей из шаблонной окрестности.

*Работы выполнялись в рамках проекта ISTC B-1486.

– Дистанционные преобразования на основе решения уравнения эйконала, где расстояние находится из конечных разностей первого или второго порядков расстояний соседних вокселей.
– Евклидово дистанционное преобразование в квадрате, в котором вместо расстояния до ближайшей точки фона хранится квадрат этого расстояния, что позволяет воспользоваться некоторыми преимуществами.

По схеме обхода классификация выглядит следующим образом:
– Последовательные, в которых обход начинается из одного угла и идет в направлении противоположного угла. Обычно требуют несколько проходов по всему объему данных.
– Волновые, в которых вычисление расстояний начинается от границы в порядке увеличения расстояния. Если требуется, вычисления можно остановить на определенном значении расстояния. Это свойство волновой схемы дает ряд преимуществ, например, для вычисления морфологических операторов, построения множеств уровня.

Дистанционные преобразования Чамфера

В дистанционных преобразованиях Чамфера (ДПЧ) [4], [5] шаблон (рис. 1) центрирован относительно каждого вокселя и расстояние вычисляется как минимум среди сумм расстояний соседних вокселей и соответствующего шаблонного веса. Для преобразований этого вида разработаны обе схемы обхода (последовательная и волновая).

	F		f	
f	E	d	e	f
	D	e	d	
f	E	d	e	f
	F		f	

z-2

f	e	d	e	f
e	c	b	c	e
d	b	a	b	d
e	c	b	c	e
f	e	d	e	f

z-1

		d		
d	b	a	b	d
	a	0		

z

Рисунок 1 – Шаблон расстояний для ДП Чамфера для прямого обхода

В последовательной схеме используются два прохода: прямой и обратный. На первом проходе ищутся минимальные расстояния до нижней границы изображения, на обратном – до верхней. Далее результаты обоих проходов объединяются и получается результат.

Таблица 1 – Веса, используемые для разных метрик и размеров структурного элемента

Метрика, размер структурного элемента	a	b	c	d	e	f
Городская	1					
Шахматная	1	1				
Евклидова, 3^3	1	$\sqrt{2}$	$\sqrt{3}$			
Квази-Евклидова, 5^3	1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{5}$	$\sqrt{6}$	3

В волновой схеме воксели обрабатываются в порядке увеличения расстояния. Чтобы гарантировать правильный порядок обработки, обработанные воксели помещаются в приоритетную очередь. На каждом шаге достается воксель с головы очереди и для всех необработанных соседних вокселей вычисляется расстояние по заданному шаблону и они вставляются в хвост очереди. Алгоритм заканчивает работу, когда

очередь пуста. Асимптотическая сложность такой схемы порядка $O(N \log N)$, что хуже, чем в последовательной схеме ($O(N)$). Однако, при специальных ограничениях на величину необходимого расстояния, такой подход оправдывает себя и работает значительно быстрее, чем схема с просмотром всех вокселей.

Векторные дистанционные преобразования

ДП Чамфера обладают невысокой точностью, что особенно проявляется на больших расстояниях от поверхности объекта. Эту проблему помогают решить векторные ДП (ВДП) [6], в которых для каждого вокселя хранится вектор ближайших точек границы.

Рассмотрим ВДП [7]. Изначально вокселям объекта приписывается расстояние 0, остальным – бесконечность. Всего требуется 8 обходов по всему объему данных (4 прямых и 4 обратных) (рис. 2). При каждом обходе обновляется вектор ближайших точек границы, используя вектора соседей.

Существуют также другие шаблоны и схемы обхода для разных метрик. Некоторые позволяют делать обходы независимо друг от друга, что дает возможность распараллелить процесс вычисления карты расстояний.

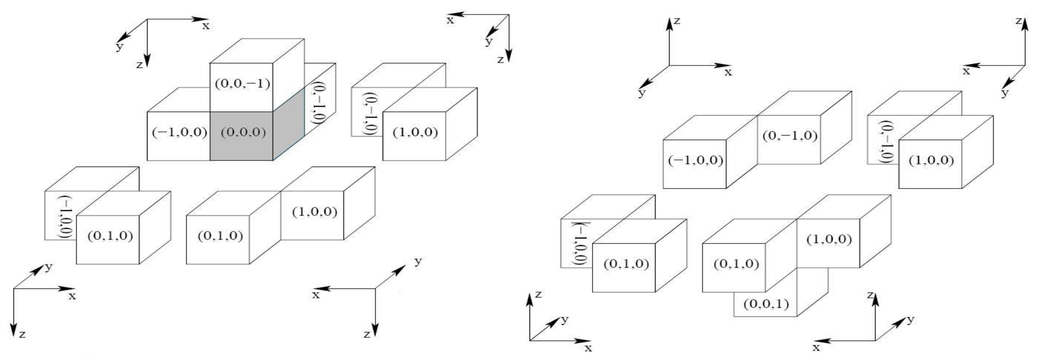


Рисунок 2 – Шаблоны для векторных ДП для прямого (слева) и обратного (справа) проходов

Быстрый маршевый метод и его вариации

Пусть у нас имеется множество Ω, $\partial\Omega$ – его граница. Уравнением эйконала называется уравнение в частных производных вида:

$$\begin{cases} \|\nabla T(x)\| F(x) = 1, x \in \Omega \\ T | x \in \partial\Omega = 0 \end{cases}, \qquad (1)$$

где ∇ означает градиент, $\|.\|$ – какую-то метрику, $F(x)$ – скорость движения точки (функция с положительными значениями. $T(x)$ называется временем прибытия и имеет следующий физический смысл: это минимальное время, за которое граница достигает точки x. Если положить $F(x)=1$, то значение $T(x)$ будет не что иное, как расстояние до ближайшей точки границы множества.

Идея алгоритма [8] заключается в том, что мы расширяем границу с постоянной единичной скоростью в направлении нормали к ней. По своей сути это метод с волновой схемой обхода и его структура почти идентична алгоритму Дейкстры нахождения кратчайшего пути. В классическом алгоритме быстрой маршировки расстояния

пересчитываются с помощью формулы конечных разностей первого порядка. Для нахождения вокселя с минимальным значением расстояния используется приоритетная очередь. Сложность управления приоритетной очередью порядка $O(N \log N)$, где N – число вокселей. Как отмечалось выше, в целом этот алгоритм, как и все алгоритмы с волновой схемой, работает медленнее, чем последовательные ДП.

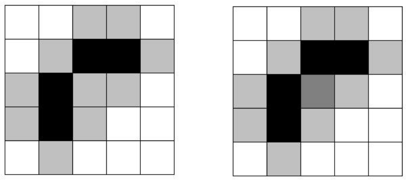


Рисунок 3 – Пример работы быстрого маршевого метода. Изначально просматриваются все воксели, смежные с граничными (черные клетки). Затем выбирается клетка с наименьшим значением расстояния и помечается как просмотренная

Также для пересчета расстояний можно использовать формулу конечных разностей второго порядка, что увеличивает точность вычислений.

Яцив и др. [9] заменили кучу массивом связанных списков. Операции добавления и извлечения выполняются за $O(1)$, но из-за того, что списки несортированные, возникают некоторые погрешности. В некоторых случаях они приемлемы, учитывая экономию времени. Но на сложных объектах ошибки становятся существенными.

Евклидово дистанционное преобразование в квадрате

Этот алгоритм был разработан Мейстером, Рординком и Хесселинком [10]. Основная идея – использование дистанционных карт меньших размерностей.

Опишем алгоритм для двухмерного изображения. Трехмерный случай аналогичен.

По определению:

$$I_d(x, y) = \min\{(x-i)^2 + (y-j)^2\}, I(i, j) \in O, \tag{2}$$

где $I(i, j)$ – пиксель с координатами (i, j).

Формула (2) может быть переписана следующим образом:

$$I_d(x, y) = \min_i\{(x-i)^2 + G(i, y)^2\}, \tag{3}$$

где

$$g(i, y) = \min_j\{|y-j|\}, I(i, j) \in O. \tag{4}$$

Алгоритм состоит из двух фаз. Первая фаза использует 2 прохода для вычисления дистанционного преобразования по столбцам (функция G). Это несложно сделать с помощью динамического программирования за линейное время.

Во второй фазе идет вычисление строка за строкой для всех x с фиксированной y-координатой. Поэтому можем ввести следующее обозначение $g(i) = G(i, y)$.

Затем:

$$d(x, y) = \min_i f(x, i), \tag{5}$$

где

$$f(x, i) = (x-i)^2 + g(i)^2. \tag{6}$$

Полезно рассмотреть геометрическую интерпретацию формулы (6). График $f(x,i)$ – это парабола с вершиной $(i, g(i))$. И (5) мы можем интерпретировать как нижнюю границу семейства парабол $f(x,i)$ (рис. 4).

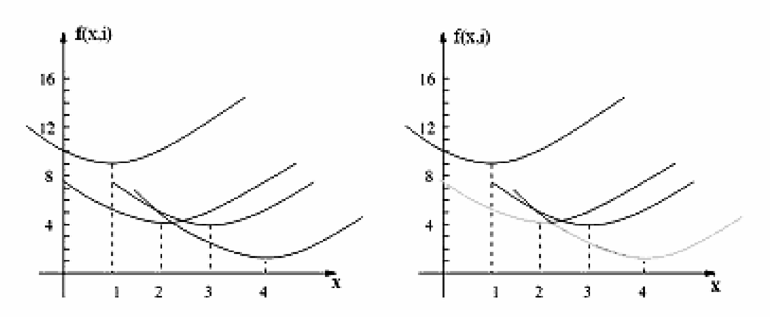


Рисунок 4 – Нижняя граница семейства парабол

Чтобы найти нижнюю границу, все параболы помещаются в стек. Когда рассматривается очередная парабола, она выталкивает параболы из стека, которые ее выше. Дополнительно хранится информация, на каком участке произошло выталкивание. Затем эта парабола заносится в стек. В итоге легко можно определить нижнюю границу.

Так как вычисления для столбцов (строк) независимы, алгоритм может быть распараллелен.

Сравнение

Алгоритмы сравнивались на нескольких медицинских изображениях. Первый тест выполнялся на объектах простой формы: точка в центре области и лимфоузлы, полученные с изображений компьютерной томографии (КТ). Второй – для изображений сложной формы: кость с полостью (КТ), щитовидная железа (ультразвуковое изображение, УЗИ), опухоль забрюшинного пространства (КТ). Третий – для сложных ветвящихся объектов: сгенерированное дерево с множеством сегментов небольшой толщины, ангиографическое изображение сосудов (КТ), клеточный конгломерат, полученный методами кофокальной микроскопии.

Максимальная ошибка (МО) вычислялась по формуле:

$$ME(I) = \max(|I(x) - I_{etalon}(x)|) \quad \forall x \in I\,, \tag{7}$$

где I – построенная карта расстояний, I_{etalon} – эталонная карта.

Средняя ошибка вычислялась следующим образом:

$$AE(I) = \frac{\sum_{x \in I} |I(x) - I_{etalon}(x)|}{N}\,, \tag{8}$$

где N – общее число вокселей.

Тестирование проводилось на двуядерном процессоре Core 2 Duo 1.66 ГГц.

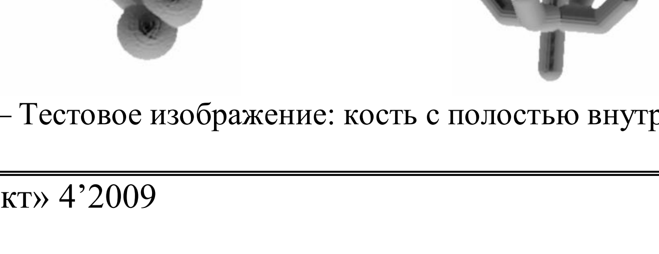


Рисунок 5 – Тестовое изображение: кость с полостью внутри и дерево



Рисунок 6 – Слои тестовых медицинских изображений: компьютерной томографии, ультразвукового исследования

Результаты работы алгоритмов представлены в табл. 2.

Таблица 2 – Сравнительная таблица алгоритмов (СО – средняя ошибка, МО – максимальная ошибка, В – время выполнения)

Алгоритм	1 тест			2 тест			3 тест		
	СО	МО	В	СО	МО	В	СО	МО	В
ВДП	0	0	7.10	0.003	0.097	9.10	0.02	0.31	9.35
ДПЧ (городская м-ка)	63.11	149.7	5.52	17.38	93.32	7.74	11.30	76.93	7.76
ДПЧ (шахматная м-ка)	15.93	49.58	6.91	10.39	35.79	8.67	4.16	22.38	8.35
ДПЧ (евклидова м-ка, 3^3)	8.48	15.03	11.70	2.51	15.84	12.33	2.04	13.14	10.85
ДПЧ (евклидова м-ка, 5^3)	2.15	6.31	35.56	0.44	3.69	48.28	0.59	4.50	32.74
Упрощенный БММ	0.37	0.55	125.8	0.04	0.27	116.2	0.06	0.93	130.23
Упрощенный БММ, $\tau = 3$	0.01	0.01	0.22	0.026	0.01	3.32	0.05	0.055	3.48
ДПЧ (евклидова м-ка, 3^3), $\tau = 3$	0.21	0.41	0.21	0.27	0.65	3.48	0.29	0.71	3.71
ЕДП2	0	0	5.80	0	0	8.17	0	0	9.05

Некоторые результаты сравнений представлены на рис. 7 и рис. 8.

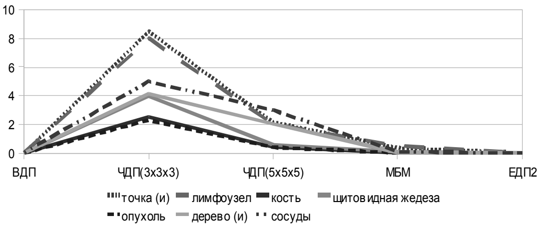


Рисунок 7 – Сравнение алгоритмов – средняя ошибка

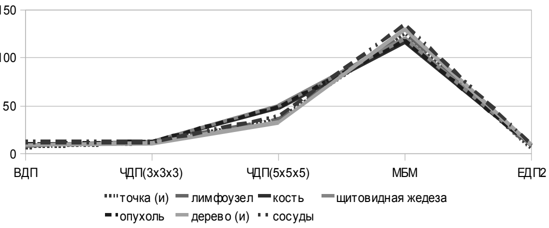


Рисунок 8 – Сравнение алгоритмов – скорость работы (с)

На некоторых тестах самыми быстрыми оказались ДП Чамфера. Они также просты в реализации, время их работы предсказуемо. ДПЧ с маской размерами 5×5×5 оказалось точнее ДПЧ с маской размерами 3×3×3, но время в 3 – 4 раза выше. ДПЧ могут применяться в областях, где важнее скорость, чем точность (например, при ускорении метода бегущего луча).

Векторное ДП также было быстрым на некоторых тестах, хотя его точность намного выше, чем у ДП Чамфера. Тем не менее, этот метод не является абсолютно точным – на большом удалении от границы объекта начинают появляться ошибки.

Также из-за необходимости хранения векторов ближайших точек алгоритм использует много памяти. Он может применяться в областях, где важны и скорость, и точность (обработка столкновений, морфинг).

Упрощенный быстрый маршевый метод из-за своей асимптотической сложности значительно проигрывает остальным методам, но все же обладает очень хорошей точностью на несложных объектах. Благодаря волновой схеме обхода данных, его можно использовать там, где интересуемое расстояние ограничено (например, морфологические операторы) и не требуется полный проход по данным. В этих случаях он быстрее алгоритмов с последовательной схемой прохода.

Выводы

Самым точным преобразованием для всех классов объектов оказалось ЕДП2. Однако из-за своей сложности этот алгоритм оказался не самым быстрым – на объектах первой и цельных объектов второй группы (щитовидная железа и опухоли) быстрее оказалось ВДП и на более сложных объектах второй группы (кости и опухоли сложной формы, сосуды) – ЧДП. ЧДП с маской размерами 3×3×3 было быстрее, чем ЧДП с маской 5×5×5, но построенная карта расстояний имеет существенно худшее качество. Результаты работы МБМ самые медленные, но благодаря его волновой схеме обхода этот метод можно использовать для анализа очень сложных объектов, таких как сосудистые и нервные сети. Для исследования небольших фрагментов изображений время работы алгоритма удовлетворительное и позволяет включать его в автоматизированные системы морфометрического анализа.

Литература

1. Rosenfeld A. Sequential operations in digital picture processing / A. Rosenfeld, J.L. Pfaltz // Journal of the ACM. – 1966. – № 13 (4). – P. 471-494.
2. Serra J. Image Analysis and Mathematical Morphology / J. Serra // Academic Press. – 1982.
3. Borgefors G. Computing skeletons in three dimensions / G. Borgefors, I. Nyström, G. Sanniti di Baja // Pattern Recognition. – 1999. – № 32(7). – P. 1225-1236.
4. Cuisenaire O. Applications of the region growing euclidean distance transform: Anisotropy and skeletons / O. Cuisenaire, B. Macq // In Proc. ICIP. – 2000. – V. I. – P. 200-203.
5. Svensson S. Distance transforms in 3D using four different weights / S. Svensson, G. Borgefors // Pattern Recognition Letters. – 2002. – V. 23. – P. 1407-1418.
6. Danielsson P.-E. Euclidean distance mapping / P.-E. Danielsson // Computer Graphics and Image Processing. – 1980. – V. 14. – P. 227-248.
7. Satherley R. Vector-city vector distance transform / R. Satherley, M.W. Jones // Computer Vision and Image Understanding. – 2001. – V. 82(3). – P. 238-254.
8. Sethian J.A. Fast marching methods / J.A. Sethian // SIAM Review. – 1999. – V. 41 (2). – P. 199-235.
9. Yatziv L. O (n) implementation of the fast marching algorithm / L. Yatziv, A. Bartesaghi, G. Sapiro // IMA Preprint Series. – 2005. – № 2021.
10. Meijster A. A General Algorithm for Computing Distance Transforms in Linear Time / A. Meijster, J.B.T.M. Roerdink, W.H. Hesselink // Computational Imaging and Vision. Mathematical Morphology and its Applications to Image and Signal Processing. – 2000. – V. 18. – P. 331-340.

Д.А. Гончаров, А.М. Недзьведь, С.В. Абламейко

Аналіз методів побудови об'ємних дистанційних карт

У статті розглянуті і зіставлені один з одним за декількома аспектами (точність, швидкість, простота реалізації, можливість розпаралелювання) методи побудови тривимірних дистанційних карт, в результаті чого можна з певною долею впевненості дати відповідь на питання, в якому випадку краще використовувати той або інший метод.

D. Goncharov, A. Nedzved, S. Ablameyko

Analysis Methods for Constructing Volumetric Distance Maps

Methods of distance map construction were examined and compared with each other in several aspects (accuracy, speed, ease of implementation, the possibility of paralleling) techniques for constructing three-dimensional distance. With some confidence you can give an answer to a question, in which case it is better to use one or another method.

Статья поступила в редакцию 19.08.2009.